Deep Learning (CSE641/ECE555)
Quiz-2 (15 Marks) (Duration 60 min) %

Name ......
Roll No. ......

Question 1-12 [1 Marks], Question 13 [3 Marks]

1. In an LSTM cell, which computations occur inside the three gates (input, forget, and output) for the given variables: cell state c_t, hidden state h_t, and input x_t?

   (a) Forget gate: $f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$, Input gate: $i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$, Output gate: $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

   (b) Forget gate: $f_t = \tanh(W_f c_t + b_f)$, Input gate: $i_t = \sigma(W_i x_t + b_i)$, Output gate: $o_t = \mathrm{ReLU}(W_o h_t + b_o)$

   (c) Forget gate: $f_t = \sigma(W_f x_t + b_f)$, Input gate: $i_t = \sigma(W_i h_{t-1} + b_i)$, Output gate: $o_t = \tanh(W_o c_t + b_o)$

   (d) Forget gate: $f_t = \mathrm{softmax}(W_f[h_{t-1}, x_t] + b_f)$, Input gate: $i_t = \mathrm{softmax}(W_i[h_{t-1}, x_t] + b_i)$, Output gate: $o_t = \mathrm{softmax}(W_o[h_{t-1}, x_t] + b_o)$

   A

2. Given the following values for an LSTM cell at time step t:

$$h_{t-1} = 0.5, \quad x_t = 0.3, \quad W_f = 1.2, \quad W_i = 0.8,$$
$$W_o = 1.5, \quad b_f = -0.1, \quad b_i = 0.2, \quad b_o = 0.05$$

   Compute the output of the forget gate f_t (rounded to 2 decimal places).

   (a) 0.55

   (b) 0.82

   (c) 0.70

   (d) 0.91

   C

3. What is the primary theoretical advantage of multi-head attention over single-head attention in transformer models?

   (a) It reduces the computational complexity of the self-attention mechanism

   (b) It allows the model to attend to information from different representation subspaces simultaneously

   (c) It eliminates the need for feed-forward networks in transformer architectures

   (d) It provides a more efficient alternative to recurrent neural networks

   B

4. Which of the following statements about the multi-head self-attention mechanism is correct?

   (a) It requires sequential processing of each head, making it much more expensive than single-head attention.

   (b) It has a similar cost to single-head attention since each head operates on a lower-dimensional representation.

   (c) It duplicates full-dimensional computation for each head, making it significantly more expensive.

   (d) It removes the need for linear projections, reducing computational cost.

   B: In multi-head self-attention, the input is projected into multiple subspaces using learned matrices, where each head operates on a lower-dimensional representation (e.g., $d_k = \frac{d_{\text{model}}}{h}$ per head). Despite having multiple heads, the overall computational cost remains similar to that of a single-head attention mechanism operating in full-dimensional space because the reduced dimensionality per head balances out the cost of having multiple heads.

5. Apart from the well-known scaled dot-product attention (SDPA) method, how else can the attention score be computed using a kernel function?

(a) $\alpha_i = K(q, k_i)$ using a similarity kernel $K(q, k)$.

(b) $\alpha_i = \frac{K(q,k_i)}{\sum_j K(q,k_j)}$ using a similarity kernel $K(q, k)$.

(c) $\alpha_i = K(q, v_i)$ instead of using keys k_i.

(d) $\alpha_i = K(q, k_i) \cdot v_i$ incorporating values directly.

B

6. In PyTorch, which function is used to reset the hidden state of an LSTM during training?

   (a) lstm.reset_parameters()

   (b) lstm.zero_grad()

   (c) hidden_state.detach_()

   (d) hidden_state = None

   C

7. What is the primary benefit of Layer Normalization over Batch Normalization in Seq2Seq models?

   (a) Layer Normalization is suitable for variable-length sequences

   (b) Layer Normalization requires fewer parameters than Batch Normalization

   (c) Layer Normalization is computationally less expensive

   (d) Layer Normalization removes the need of residual connection in each layer

   A

8. In PyTorch's Transformer implementation, what does torch.nn.MultiheadAttention return?

   (a) The attention scores only

   (b) The attention outputs and softmax probabilities

   (c) The output tensor and attention weights

   (d) The updated key-value cache for caching

   C

9. What is the effect of mixed-precision training using torch.cuda.amp?

   (a) Reduces memory usage and speeds up computation by using FP16 where possible

   (b) Increases model robustness by adding noise to gradients

   (c) Increases numerical precision by enforcing FP64 operations

   (d) Forces all computations to use FP32

   A

10. In self-attention, given scaled dot-product scores [2, 0, -1], what are the attention weights after softmax?

    (a) (0.88, 0.11, 0.01)

    (b) (0.5, 0.3, 0.2)

    (c) (0.7, 0.2, 0.1)

    (d) (0.9, 0.05, 0.05)

    A

11. Which mathematical operation is used to implement the gating mechanisms in a GRU? (a) Matrix addition (b) Matrix multiplication (Dot product) (c) Element-wise multiplication (Hadamard product) (d) Convolution (Discrete convolution) C

12. What is the time complexity of the self-attention mechanism in a Transformer for a sequence of length n? (a) $O(n)$ (b) $O(n \log n)$ (c) $O(n^2)$ (d) $O(1)$ C

13. Consider a simple Recurrent Neural Network (RNN) for token classification with the following definitions:

    - **Input at time step** t: $x_t \in \mathbb{R}^{n_x}$.

- **Hidden state**: $s_t \in \mathbb{R}^{n_h}$.
- **Output**: $y_t \in \mathbb{R}^{n_y}$.
- **Weight matrices**:
  - $W \in \mathbb{R}^{n_h \times n_x}$ (input-to-hidden weights),
  - $U \in \mathbb{R}^{n_h \times n_h}$ (hidden-to-hidden weights),
  - $V \in \mathbb{R}^{n_y \times n_h}$ (hidden-to-output weights).

The forward pass equations are:

$$s_t = tanh(Us_{t-1} + Wx_t), \tag{1}$$
$$y_t = Vs_t, \tag{2}$$

where $\phi(s)$ is an activation function, typically tanh.

The loss function is defined as:

$$L = \sum_t L_t = \sum_t \ell(y_t, \hat{y}_t). \tag{3}$$

Using Backpropagation Through Time (BPTT), derive the gradient of the loss function with respect to U.

Step 1: Compute the derivative of L with respect to s_t Using the chain rule:

$$\frac{\partial L}{\partial s_t} = \frac{\partial L_t}{\partial y_t}\frac{\partial y_t}{\partial s_t} + \frac{\partial L}{\partial s_{t+1}}\frac{\partial s_{t+1}}{\partial s_t}. \tag{4}$$

Since $y_t = Vs_t$, we get:

$$\frac{\partial y_t}{\partial s_t} = V. \tag{5}$$

Also, from the hidden state equation $s_t = \phi(Us_{t-1} + Wx_t)$, applying the chain rule gives:

$$\frac{\partial s_{t+1}}{\partial s_t} = U\phi'(Us_t + Wx_{t+1}). \tag{6}$$

Thus, we recursively compute:

$$\frac{\partial L}{\partial s_t} = V^T \frac{\partial L_t}{\partial y_t} + U^T \frac{\partial L}{\partial s_{t+1}} \phi'(Us_t + Wx_t). \tag{7}$$

Step 2: Compute $\frac{\partial L}{\partial U}$

Since s_t depends on U as:

$$s_t = \phi(Us_{t-1} + Wx_t), \tag{8}$$

we differentiate w.r.t. U:

$$\frac{\partial s_t}{\partial U} = \phi'(Us_{t-1} + Wx_t)s_{t-1}^T. \tag{9}$$

Thus, summing over all time steps:

$$\frac{\partial L}{\partial U} = \sum_t \frac{\partial L}{\partial s_t} s_{t-1}^T. \tag{10}$$